

Pacotes e Visibilidade de Classes

Luiz Henrique de Campos Merschmann
Departamento de Computação Aplicada
Universidade Federal de Lavras

luiz.hcm@ufla.br



Na Aula de Hoje



Modularização

Classes e métodos ajudam a **modularizar** um programa separando suas tarefas em unidades autocontidas.



Quais são as motivações para se modularizar um programa?

- ▶ Abordagem *dividir para conquistar*: torna o desenvolvimento de programas mais gerenciável, construindo programas a partir de peças mais simples e menores.
- ▶ Reutilização de software: o uso de classes e métodos existentes como blocos de construção para criar novos programas.

Pacotes em Java

Escrevemos programas Java combinando nossos métodos e classes com aqueles pré-definidos na Java API (biblioteca de classes) e em quaisquer outras bibliotecas de classes.



Classes relacionadas são agrupadas em **pacotes** de modo que possam ser **importadas** nos programas e **reutilizadas**.

```
7 string sInput;
8 int lInput, b;
9 double dInput;
10 boolean eInput = true;
11
12 while (eInput) {
13     int n = -1;
14     eInput = false;
15     getLine(cis, sInput);
16     System.out.println("?");
17     String strInput = cis.readLine();
18     lInput = strInput.length();
19     if (lInput > 0) {
20         if (strInput.charAt(0) == 'I') {
21             eInput = true;
22             continue;
23         } else if (strInput.charAt(0) == 'E') {
24             eInput = true;
25             continue;
26         } else if (strInput.charAt(0) == 'D') {
27             eInput = true;
28             continue;
29         } else if (strInput.charAt(0) == 'S') {
30             eInput = true;
31             continue;
32         } else if (strInput.charAt(0) == 'Q') {
33             eInput = true;
34             continue;
35         } else if (lInput >= 2) {
36             eInput = true;
37             continue;
38         }
39     }
40 }
```

Pacotes da Biblioteca de Classes Java



Java contém muitas classes predefinidas que estão agrupadas em pacotes. A seguir temos um subconjunto de pacotes da Java API.

O pacote `java.lang` é automaticamente importado pelo compilador em todos os programas!

Pacote	Descrição
<code>java.lang</code>	Tipos primitivos e funcionalidades básicas da linguagem
<code>java.util</code>	Coleções, modelo de eventos, data/hora, números aleatórios, etc.
<code>java.io</code>	Operações de entrada/saída
<code>java.math</code>	Aritmética de inteiros e decimais de precisão arbitrária
<code>java.nio</code>	Novo framework de entrada/saída (buffers)
<code>java.net</code>	Classes para desenvolver aplicações de rede
<code>java.security</code>	Classes e interfaces relacionadas à segurança
<code>java.sql</code>	Acesso a banco de dados
<code>java.awt</code>	Interface de usuário, desenho e imagens
<code>javax.swing</code>	Componentes de interface de usuário independente de plataforma
<code>java.applet</code>	Criação de applets

Pacotes em Java

Ao implementar uma classe podemos definir que ela pertence a um determinado pacote.

Como fazer isso?



- ▶ Inserindo uma declaração *package* antes da definição da classe:

```
package banco;  
  
class ContaBancaria{  
    //...  
}
```

```
import banco.ContaBancaria;  
  
public class ContaBancariaTeste{  
    public static void main (String[ ] args){  
        ContaBancaria minhaConta = new  
        ContaBancaria( );  
    }  
}
```

Fazendo isso, quando quisermos utilizar objetos da classe **ContaBancaria** em algum outro código, precisamos informar para a JVM que essa classe está no pacote *banco*.

Pacotes

Observações importantes para se utilizar pacotes em Java:

- ▶ Assim como uma classe em Java deve ser salva em um arquivo com o seu nome, um pacote deve ser salvo em uma pasta no sistema que possua o mesmo nome do pacote.
 - ▶ Portanto, em nosso exemplo, o arquivo *ContaBancaria.java* deve ficar na pasta chamada *banco*.
- ▶ Um pacote pode conter outros pacotes (assim como pastas podem conter subpastas).
 - ▶ Por exemplo, poderíamos ter dentro do pacote *banco* os pacotes: *nacional* e *estadual*, por exemplo.
 - ▶ Nesse caso, para usar a classe ContaBancaria, precisaríamos declarar por exemplo:
import banco.nacional.ContaBancaria;

Pacotes

Observações importantes para se utilizar pacotes em Java:

- Repare que no exemplo anterior precisamos referenciar a classe ContaBancaria com todo o nome do pacote na sua frente (nome qualificado).

import banco.nacional.ContaBancaria;

- No entanto, nesse caso, poderíamos usar também o comando:

import banco.nacional.*;

Isso indica que estamos importando **todas** as classes contidas no pacote *banco.nacional*.

Cuidado!



Com o comando **import banco.*** só importaremos as classes diretamente vinculadas ao pacote *banco* (na pasta *banco*), ou seja, não conseguiremos importar as classes contidas em subpacotes (subpastas de *banco*, como por ex., *nacional*).

Pacotes

Observações importantes para se utilizar pacotes em Java:

- ▶ Existe uma outra maneira de trabalhar com classes contidas em pacotes sem usar o comando **import**.
 - ▶ Basta você qualificar totalmente o nome da classe sempre que precisar dele no seu código.
- ▶ Por exemplo:

```
public class ContaBancariaTeste{  
    public static void main (String[ ] args){  
        banco.nacional.ContaBancaria minhaConta = new  
        banco.nacional.ContaBancaria( );  
    }  
}
```

Pacotes

Imagine a seguinte situação...

Duas equipes de uma empresa que trabalham em projetos distintos criaram classes para serem reutilizadas em outros códigos.

Equipe A: está desenvolvendo um sistema para uma loja de instrumentos musicais e, por isso, criou uma classe chamada Teclado.



Equipe B: está desenvolvendo um sistema para uma loja de informática e, por isso, também criou uma classe denominada Teclado.

Pacotes

Como evitar o conflito de nomes para essas classes criadas pelas equipes?

- ▶ Se cada projeto for colocado dentro de um **pacote** diferente, ainda que as classes tenham nomes iguais, o sistema irá diferenciá-las pelo nome do pacote.
- ▶ Portanto, poderiam ser definidos os pacotes *informatica* e *musica*.
- ▶ Por exemplo, se precisarmos criar um objeto teclado musical, basta importar a classe Teclado do pacote *musica*:
import musica.Teclado;

E se os dois pacotes forem importados em algum código?



Nesse caso, faz-se necessário utilizar o nome qualificado da classe no código. Por exemplo:
musica.Teclado inst = **new** *musica.Teclado()*;

Pacotes

Convenção de nomeação de pacotes



- ▶ Existe uma convenção (padrão) para nomear pacotes em Java.
- ▶ Apesar de você não ser obrigado a seguir essa convenção, para facilitar a sua vida e a das pessoas ao seu redor, utilize-a.
- ▶ A ideia é garantir **nomes** de pacotes **únicos**.

Portanto:

- ▶ Use letras **minúsculas** para os nomes de pacotes (evita conflito com os nomes das classes).
- ▶ Comece-o com o nome de domínio na internet da sua empresa ou instituição na ordem inversa.

Exemplo: Domínio: dac.ufla.br → Pacote: *br.ufla.dac.meupacote*

- ▶ Você ainda pode especificar partes adicionais nesse nome.
Exemplo: Se a classe a ser inserida no pacote é de um projeto específico, digamos, poo, você inserir essa informação no nome:
br.ufla.dac.poo.meupacote

Pacotes x Visibilidade de Classes

Visibilidade de Classes



Já vimos que é possível definir a **visibilidade** de atributos e métodos.

Mas e para uma classe? Será que podemos definir sua visibilidade também?

Sim, podemos! Uma classe pode ser **pública**, por exemplo.

```
public class ContaBancaria{  
    ...  
}
```

Nesse caso, a classe é visível por **qualquer outra classe** do sistema. Apenas classes *public* podem ser reutilizadas fora do pacote em que estão declaradas!

Pacotes x Visibilidade de Classes

Visibilidade de Classes

Mas e se eu não colocar nada? Qual será a visibilidade da classe?



```
class ContaBancaria{  
    ...  
}
```

Ela será acessível por qualquer outra que esteja dentro do **mesmo pacote!**

- ▶ Mas e quando eu não inseri minhas classes em pacote algum?
 - ▶ Se nenhuma instrução *package* for fornecida em arquivos de código-fonte Java, **todas as classes** declaradas nesses arquivos serão inseridas no chamado **pacote padrão** e só serão acessíveis a outras classes no pacote padrão que estão localizadas no **mesmo diretório**¹.

¹O mesmo vale para métodos e atributos que você deixou sem um modificador de visibilidade.

Visibilidade de Classes



Uma classe também pode usar os modificadores de visibilidade **protected** ou **private**.

Como assim?



KEEP CALM
AND
AGUARDE: CENAS DOS
PRÓXIMOS
CAPÍTULOS

Perguntas?

